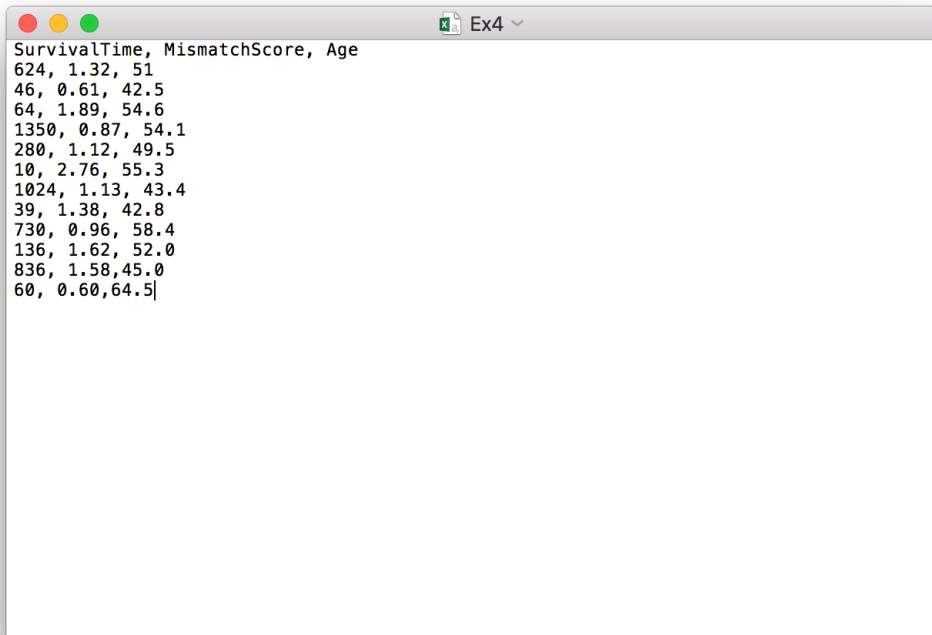


Problem 4:



SurvivalTime	MismatchScore	Age
624	1.32	51
46	0.61	42.5
64	1.89	54.6
1350	0.87	54.1
280	1.12	49.5
10	2.76	55.3
1024	1.13	43.4
39	1.38	42.8
730	0.96	58.4
136	1.62	52.0
836	1.58	45.0
60	0.60	64.5

Below is the R program used for this problem.

```
# read csv file
mydata <- read.csv("/Users/macbookpro/Desktop/Ex4.csv")
x1<- mydata$MismatchScore;
x2<- mydata$Age;
t <- mydata$SurvivalTime;
y <- log(t);

#fit log model
fit <- lm(y ~ x1 + x2)

#Results of the model
summary(fit)
```

```
> # read csv file
> mydata <- read.csv("/Users/macbookpro/Desktop/Ex4.csv")
>
> x1<- mydata$MismatchScore;
> x2<- mydata$Age;
> t <- mydata$SurvivalTime;
> y <- log(t);
>
> #fit log model
> fit <- lm(y~x1 + x2)
>
> #Results of the model
> summary(fit)
```

Call:

lm(formula = y ~ x1 + x2)

Residuals:

	Min	1Q	Median	3Q	Max
	-2.4130	-1.2578	0.1148	1.2471	1.6651

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	8.02139	3.72230	2.155	0.0596 .
x1	-1.14911	0.79014	-1.454	0.1798
x2	-0.02538	0.06959	-0.365	0.7237

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.578 on 9 degrees of freedom

Multiple R-squared: 0.2007, Adjusted R-squared: 0.02308

F-statistic: 1.13 on 2 and 9 DF, p-value: 0.3649

>

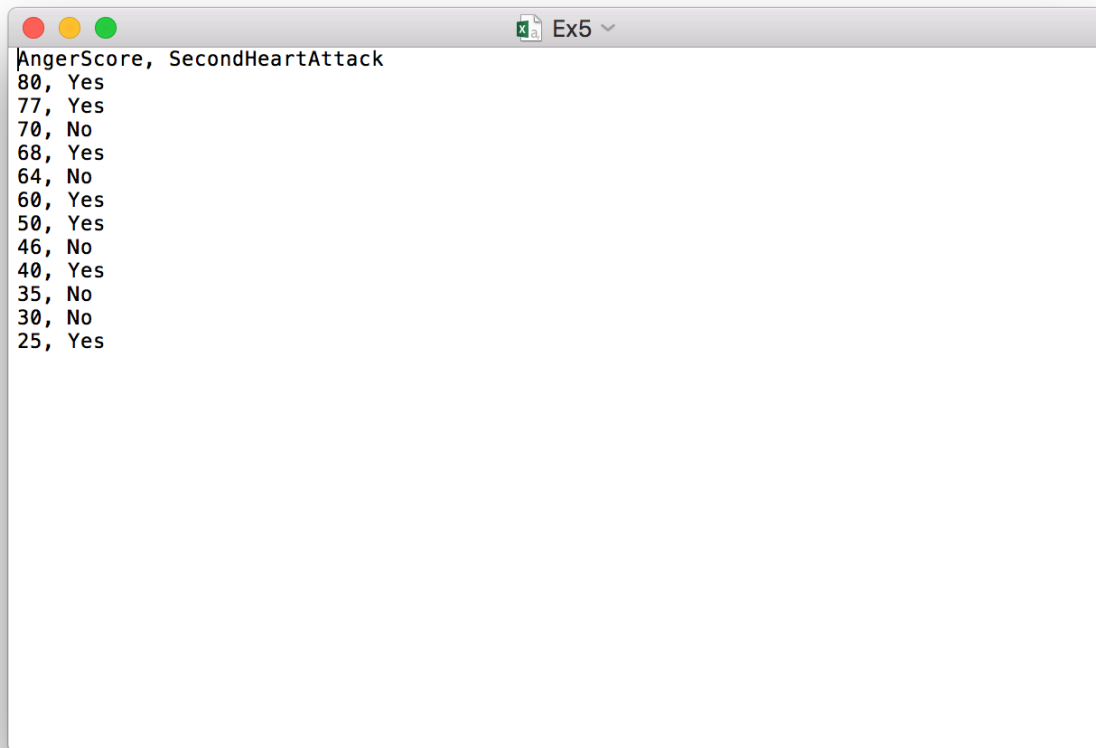
- (a) Let the dependent variable be the logarithm of Survival time. Fit a multiple linear regression on the independent variables of Mismatch score and Age.

Solution: Based on the program, we can see the alpha equals 8.02139. The estimate of Mismatch score (x1) equals -1.14911. The estimate of Age (x2) equals -0.02538.

- (b) Compute an estimate of the variance of the error term.

Solution: The estimate of the variance of the error term will be the square of 1.578 and that will be 2.4901.

Problem 5:



AngerScore	SecondHeartAttack
80	Yes
77	Yes
70	No
68	Yes
64	No
60	Yes
50	Yes
46	No
40	Yes
35	No
30	No
25	Yes

Below is the R program used for this problem.

```
# read csv file
mydata <- read.csv("/Users/macbookpro/Desktop/Ex5.csv")
y <- mydata$SecondHeartAttack;
x <- mydata$AngerScore;

# change "YES" or "NO" into booleans
for (i in mydata$SecondHeartAttack){
  if (i == "Yes"){
    i <- 1
  }
  else{
    i <- 0
  }
}
```

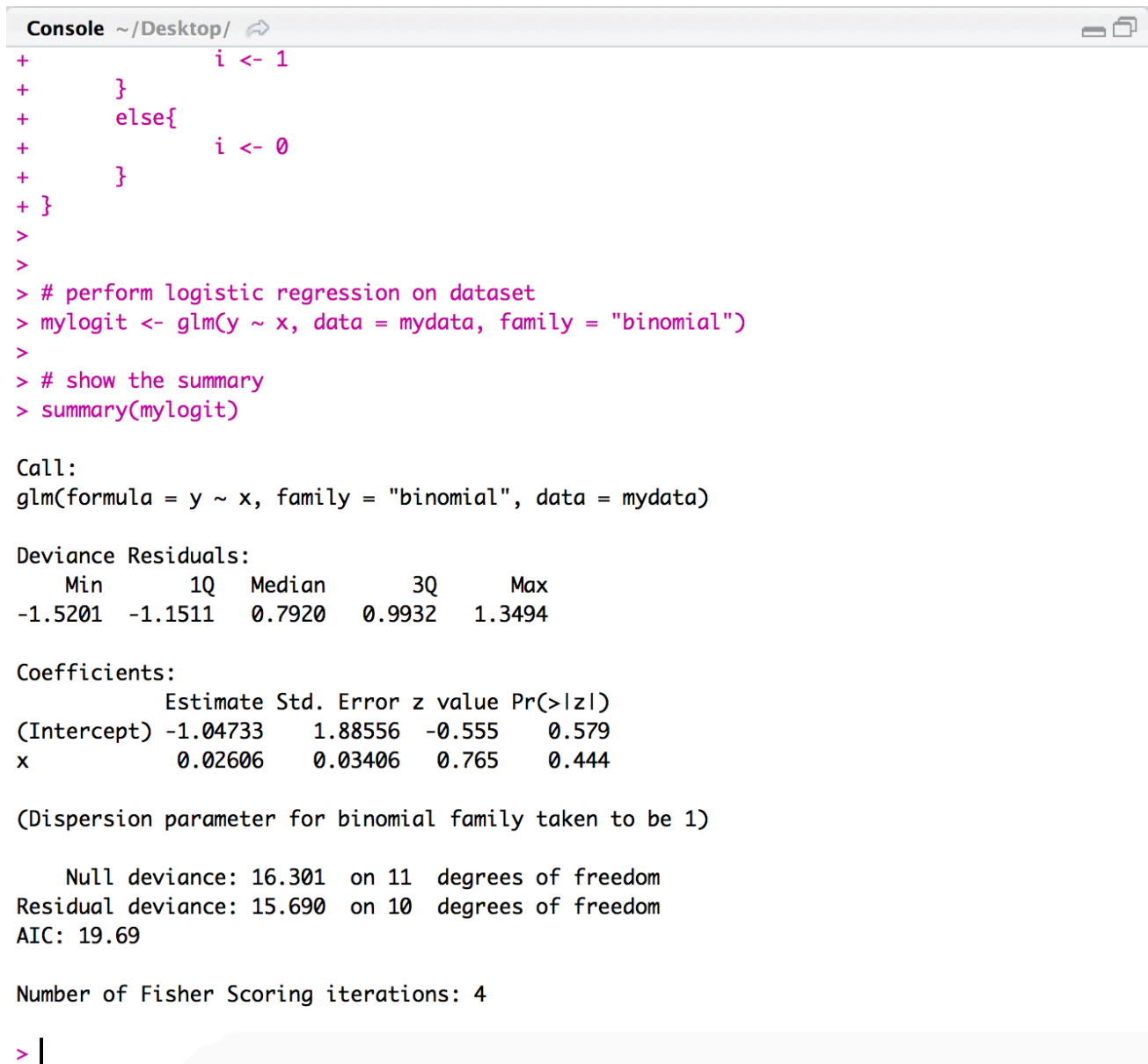
```

}

# perform logistic regression on dataset
mylogit <- glm(y ~ x, data = mydata, family = "binomial")

# show the summary
summary(mylogit)

```



The screenshot shows a R console window with the following content:

```

Console ~/Desktop/
+           i <- 1
+       }
+   else{
+       i <- 0
+   }
+ }
>
>
> # perform logistic regression on dataset
> mylogit <- glm(y ~ x, data = mydata, family = "binomial")
>
> # show the summary
> summary(mylogit)

```

Call:

```
glm(formula = y ~ x, family = "binomial", data = mydata)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.5201	-1.1511	0.7920	0.9932	1.3494

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.04733	1.88556	-0.555	0.579
x	0.02606	0.03406	0.765	0.444

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 16.301 on 11 degrees of freedom
 Residual deviance: 15.690 on 10 degrees of freedom
 AIC: 19.69

Number of Fisher Scoring iterations: 4

> |

(a) Explain how the relationship between a second heart attack and one's anger score can be analyzed via a logistic regression model.

Solution: Based on the dataset, the values of "Second Heart Attack" contain 'Yes' and 'No'. They can be categorized as two categories. So to some extent, it is a 'Binary Classification' problem. So it can be solved via logistic regression.

(b) Using a software package of your choice, estimate parameters for this model (for example, in Matlab to fit a logistic model consider the command 'glmfit').

(c) Estimate the probability that a heart attack patient with an anger score of 55 will have a second heart attack within 5 years.

Solution: Based on the program, $\beta_0 = -1.04733$, $\beta_1 = -0.02606$. So with $x = 55$, the answer will be 0.99999. So the patient will have a heart attack within 5 years.

Problem 5:

```
# Read data
mydata <- read.csv("/Users/macbookpro/Desktop/IE 529 HW3/Data-
HW3/PCAdata.csv",header = FALSE)

# Compute the SVD (singular value decomposition)
SVD <- svd(mydata)

# Standardized the data
for (i in 1:3){
  transpose[,i] = transpose[,i] - mean(transpose[,i])
}

# find eigenvalues and eigenvectors; Covariance
my_cov <- cov(transpose)
my_eigen <- eigen(my_cov)
```

(a) For this data set compute the SVD (singular value decomposition) of the original matrix, and using this SVD discuss the expected results of performing a PCA on this data.

(b) Compute the PCA: First compute the mean(s) for the data, and subtract from the original data; second compute the covariance matrix including the scaling $1/(n - 1)$; third compute an eigenvalue decomposition and sort both the eigenvalues and eigenvectors in descending order.

(c) Plot and discuss the principal components. Discuss how this process and results might differ from a direct SVD of the de-biased, scaled data.

The covariance matrix C is given by $C = X^T X / (n-1)$. It is a symmetric matrix so it can be diagonalized as: $C = V L V^T$, where V is a matrix of eigenvectors (each column is an eigenvector) and L is a diagonal matrix with eigenvalues *lamda* in the decreasing order on the diagonal.

With Singular Decomposition of X , we can get $X = U S V^T$, where S is the diagonal matrix of singular values s_i . So we can use some matrix multiplication rules to get

$$C = V S U^T U S V^T / (n - 1) = V \frac{S^2}{n - 1} V^T,$$

With the eigenvalue decomposition, $C = V L V^T$, so we can obtain *lamda* = $s^2/(n-1)$. So we can know why this process might differ from SVD.